# Fișa 05 C# – tipuri definite de utilizator – struct, enum

## Tipul struct:

* 1. Modul de folosire este asemănător cu cel din C++;
  2. Se folosesc pentru a construi mici pachete de variabile care au legătură între ele: datele unui elev, datele unui obiect de inventar etc. Exemplu:

public struct elev

{

public string nume;

public double medie;

}

* 1. Pot conține constructori, constante, metode etc. (la fel ca o clasă), dar nu pot moșteni alte structuri sau clase.
  2. Câmpurile din structură nu pot fi inițializate decât dacă sunt declarate ca **static** sau **const**.
  3. Sunt de tip valoare (nu de tip referință) => se pot inițializa direct, fără a folosi **new**, câte un câmp la un moment dat; se copiază în întregime când se face o atribuire. Exemplu:

elev e1, e2;

e1.nume = "Pop Ioan";

e1.medie = 9.66;

e2 = e1; //elevul e2 va conține aceleași date ca e1

* 1. Se creează și se accesează mai rapid decât clasele sau vectorii, deci sunt recomandate când trebuie prelucrate blocuri de date simple, dar numeroase.

1. **Exemplu**:

class Program

{

public struct elev

{

public string nume;

public double medie;

}

static void Main(string[] args)

{

int n; //numărul de elevi

//citire număr de elevi

Console.Write("\introduceți numărul de elevi : ");

n = int.Parse( Console.ReadLine() );

//definire clasa

elev[] clasa = new elev[n];

//citire elevi

for (int i = 0; i < n; i++)

{

Console.Write("\introduceți numele si media elevului " + (i + 1) + " : \n");

clasa[i].nume = Console.ReadLine();

clasa[i].medie = double.Parse( Console.ReadLine() );

}

//afișare elevi

Console.Write("\nElevii din clasa sunt : ");

for (int i = 0; i < n; i++)

Console.Write("\nnumele: " + clasa[i].nume.PadRight(10) + "medie: " + clasa[i].medie);

//gata

Console.WriteLine("\n");

}

}

1. **Exercițiu**: realizați o aplicație de tip consolă care citește informații despre ***n*** elevi dintr-o clasă. Pentru fiecare elev se cunoaște numele și media. Se cere:
   1. să se afișeze media pe clasă;
   2. să se afișeze numele și media elevilor, elevii fiind ordonați descrescător după medie;
   3. să se afișeze numele și media elevilor, elevii fiind ordonați crescător după nume.

*Observații:*

* + - *compararea a două valori de tip string se face cu funcția string.Compare( s1, s2 ) care va da ca rezultat un număr pozitiv dacă s1 > s2, 0 dacă s1 == s2 sau un număr negativ dacă s1 < s2 .*

## Tipul enum:

Se folosește pentru a crea grupuri de constante consecutive. De exemplu: „luni”, „marți”, „miercuri” etc., sau „ianuarie”, „februarie”, „martie” etc., sau „unu”, „doi”, „trei” etc. Fiecărei constante din grup i se va asocia un număr întreg care va fi folosit ca și cod al șirului respectiv.

1. **Exemplu**:

class Program

{

//declarare tip enumerare

public enum ZiSaptamana {

luni = 1,

marti = 2,

miercuri = 3,

joi = 4,

vineri = 5,

sambata = 6,

duminica = 7

}

static void Main(string[] args)

{

//declarare variabile de tip enumerare

ZiSaptamana a = ZiSaptamana.luni;

ZiSaptamana b = ZiSaptamana.marti;

//afișare si transformare zi -> nr

Console.Write("\nZiua a este : " + a);

Console.Write("\n\nZiua b este : " + b + " , nr. " + (int)b );

//afișare toate zilele

Console.Write("\n\nZilele săptămânii sunt : ");

for (ZiSaptamana i = ZiSaptamana.luni; i <= ZiSaptamana.duminica; i++)

Console.Write(i + " ");

//afișare zi + nr sau transformare nr -> zi;

Console.Write("\n\nb + 1 = " + (b + 1));

Console.Write("\n\na + 8 = " + (a + 8));

Console.Write("\n\nZiua 5 este : " + (ZiSaptamana)5);

//gata

Console.WriteLine("\n\n");

}

}

1. **Exercițiu**: scrieți programe asemănătoare cu cel de mai sus în care să folosiți ca enumerări lunile anului sau numerele de la 1 la 10 scrise cu litere („unu”, „doi”, „trei” etc.).